

(19)



Europäisches Patentamt  
European Patent Office  
Office européen des brevets



(11)

EP 0 953 923 A2

(12)

## EUROPEAN PATENT APPLICATION

(43) Date of publication:  
03.11.1999 Bulletin 1999/44

(51) Int Cl.<sup>6</sup>: G06F 17/30

(21) Application number: 99303203.6

(22) Date of filing: 26.04.1999

(84) Designated Contracting States:  
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU  
MC NL PT SE  
Designated Extension States:  
AL LT LV MK RO SI

- Chiang, Luo-Jen  
Freehold, NJ 07728 (US)
- Thaisz, James E.  
Lincroft, NJ 07738 (US)
- Lou, Frank  
Bridgewater, NJ 08807 (US)

(30) Priority: 29.04.1998 US 67805

(71) Applicant: NCR INTERNATIONAL INC.  
Dayton, Ohio 45479 (US)

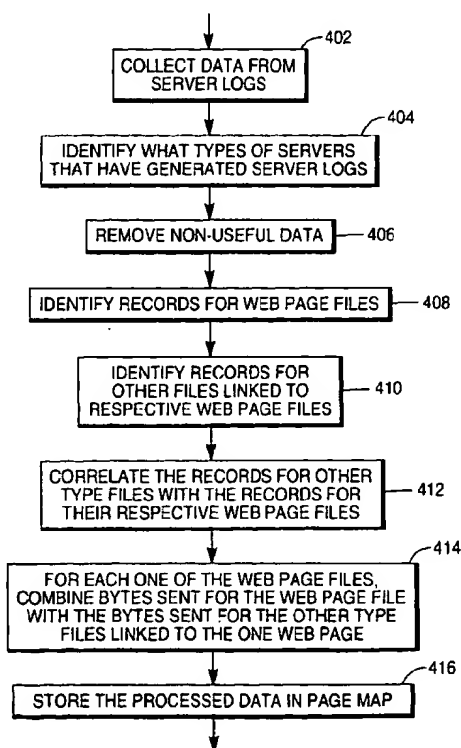
(74) Representative: Cleary, Fidelma et al  
International IP Department  
NCR Limited  
206 Marylebone Road  
London NW1 6LY (GB)

(72) Inventors:  
• Papierniak, Karen A.  
St. Paul, MN 55108 (US)

(54) **Method and apparatus for forming page map to present internet data meaningful to management and business operation**

(57) Conventionally, a web site stores Internet data indicating file access status for the files that have been accessed in response to requests from web browsers. Unfortunately, the Internet data are kept as a set of separate and non-correlated data records that are chronologically arranged according to the times at which the requests have been received and processed. Consequently, the Internet data are not arranged meaningful to management and business operation. The present invention correlates web page files (HTML, SHTML, DHTML, or CGI files) with other type files (GIF, JPEG, or AVI files), so that the Internet data can be presented in a format meaningful to management and business operation.

FIG. 4



EP 0 953 923 A2

## Description

[0001] The present invention relates generally to a method and apparatus for presenting Internet data in a format meaningful to management and business operation.

[0002] With the development in information technology and networking infrastructure, more and more business transactions are being conducted electronically over the Internet. Using the Internet to conduct business transactions are now getting so popular that it is currently well known as electronic commerce (or Internet commerce) by the industries and public. It is fair to predict that electronic commerce is having an enormous impact on the way businesses will be conducted and managed in the future. Thus, there is a great interest in studying and understanding consumers' behavior and decision process in electronic commerce environment.

[0003] Traditionally, business transactions have been conducted at business premises, and there exist methods and techniques to study consumer's behaviour and decision process for traditional business environment. For example, a retailer can display its goods in store shelves arranged in accordance with the changes of the four seasons. By observing consumers' reactions to the arrangement, the retailer can adjust the layout of the shelves to facilitate sales of its goods.

[0004] In the electronic commerce environment, a retailer or service provider typically displays information about its goods or services in a web site (which includes at least one server) via the Internet. Specifically, the server for the web site stores the information in a set of web page files, such as HTML (Hypertext Markup Language) files. In addition to containing text content, an HTML file may also contain links to other type files, such as graphic or audio files, for displaying pictures and icons and playing audio message. An HTML file may also contain links to other web page files. The other type files can be also stored on the server. By using his/her web browser, a customer (or a potential customer) can remotely navigate through the web site, gaining the information about the goods and services, or ordering selected goods or services. Unfortunately, unlike in traditional business environment, there is no reliable method in electronic commerce environment at the present time to measure the effectiveness of the layout of a web site. This is due to the difficulties in observing consumers' behaviour and analyzing consumers' decision process over the Internet.

[0005] Historically, the Internet was designed as an open structure in which the main purpose is to exchange information freely without restriction. To obtain a web page file (such as an HTML file) from a web site, a web browser first sends a request to the server for that web site. Upon receiving the request, the server retrieves the HTML file requested and send it to the web browser. Upon receiving the HTML file, the web browser displays the HTML file as a web page. If the HTML file also con-

tains links to other type files (such as graphic or audio files), the browser subsequently sends requests to the server for these files. Upon receiving the requests, the server retrieves these files and sends them to the web browser. Upon receiving these files, the browser displays pictures and icons on the web page, or executes an application to play audio files embedded in the web page. If the HTML file also contains a link to another HTML file, upon clicking (or activating) the link, the browser sends a further request to the server for the HTML file. Upon receiving the further request, the server retrieves the HTML files and sends it to the web browser. It should be noticed that browsers interact with web sites in a stateless fashion. On the Internet, a particular web site can be accessed by thousands of browsers in a random fashion. While a browser is sending a sequence of requests to a web site, it does not maintain a constant connection to that web site between any two consecutive requests. To a server, it has no control over the sequences of requests; a subsequent request may not have any logical relationship with the previous one; a sequence of requests may come from different web browsers; a request may be generated from a link embedded in an HTML file. Consequently, it is difficult to consecutively observe customers' activities and behavior in electronic commerce environment over the Internet.

[0006] Current technology provides mechanisms to record access status data (or Internet data) for web page and other type files while a sequence of requests are being received and processed by a server. However, the Internet data are kept as a set of separate and non-correlated data records that are chronologically arranged according to the times at which the requests were received and processed. Consequently, Internet data, without further processing, are not meaningful to management and business operation. In addition, since Internet data are recorded mainly for the purpose of administering web sites, they may contain redundant and erroneous data that have no use to management and business operation analysis. When Internet data are further processed by other applications (such as by data warehouse applications), these redundant and erroneous data are undesirable because they wastefully occupy storage space and may cause errors in reports or during analysis.

[0007] Moreover, Internet data may be generated by different types of servers that may use different formats to record the Internet data. In another words, Internet data generated by different types of servers are not compatible in format. This causes further problems to the utilisation of Internet data.

[0008] Therefore, there is a need for a method and apparatus to present Internet data in a format that is meaningful to management and business operation.

[0009] There is another need for a method and apparatus to present Internet data in a format that can be further efficiently utilized.

[0010] There is yet another need for a method and apparatus to filter Internet data to facilitate further analyzing process.

[0011] There is still another need for a method and apparatus to combine Internet data from multiple servers, possibly from different types of servers, into a coherent format.

[0012] The present invention meets these needs.

[0013] According to a first aspect of the present invention, there is provided a method as set out in accompanying claim 1.

[0014] According to a second aspect of the present invention, there is provided a method as set out in claim 4.

[0015] According to a third aspect of the present invention, there is provided a method as set out in accompanying claim 8.

[0016] According to another aspect of the present invention, there is provided apparatus arranged to conduct a method in accordance with any one of the first three aspects of the invention.

[0017] The present invention provides a novel method and associated apparatus for processing Internet data.

[0018] Conventionally, a web site is able to store Internet data indicating file access status for the files that have been accessed in response to requests from web browsers. Unfortunately, the Internet data are kept as a set of separate and non-correlated data records that are chronologically arranged according to the times at which the requests have been received and processed. Typically, a web page is associated with a web page file, which can further embed other type files. However, the data records indicating access status for a web page file and other type files embedded in the web page file can be scattered among multiple data records. Consequently, the Internet data are not arranged in a format meaningful to management and business operation.

[0019] The present invention converts the Internet data into a format meaningful to management and business operation. More specifically, the present invention can correlate the data record for a web page file with the data records for other type files that are embedded in the web page file.

[0020] In a broad aspect, the invention provides a method used with a server containing a plurality of web pages and logs. Each of the web pages contains a web page file and one or more other type files. Each of the logs contains data indicating access status for the web page files and other type files. The method comprises the steps of:

receiving data from the server;  
identifying data for web page files that have been accessed;  
identifying data for other type files that are respectively linked in the web page files; and  
correlating the data for the other files that are respectively linked in the accessed web page files into

the data for the web page files.

[0021] These and other features and advantages of the present invention will become apparent from the following description and accompanying drawings, which are given by way of example.

[0022] The purpose and advantages of the present invention will be apparent to those skilled in the art from the following detailed description in conjunction with the appended drawing, in which:

figure 1 shows an exemplary network system, including a novel Internet data processing computer, in accordance with the present invention;

figure 2 shows an exemplary web page associated with a web page file;

figure 3 shows exemplary data records in server logs;

figure 4 shows a flowchart illustrating the operation of forming page map shown in figure 1, in accordance with the present invention;

figure 5 shows exemplary data records stored in the page map shown in figure 1, in accordance with the present invention; and

figure 6 shows an exemplary computer system that is able run utility application shown in figure 1, in accordance with the present invention.

[0023] The present invention comprises a novel method and an associated apparatus for presenting Internet data. The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the preferred embodiment(s) will be readily apparent to those skilled in the art, and the principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the invention. Thus, the present invention is not intended to be limited to the embodiment(s) shown, but is to be accorded with the broadest scope consistent with the principles and features disclosed herein.

[0024] Referring to figure 1, there is shown an exemplary network system 100 including Internet 105 and Intranet (or LAN - Local Area Network) 107, in accordance with the present invention.

[0025] Connected to Internet 105 are four servers (102<sub>1</sub>, 102<sub>2</sub>, 102<sub>3</sub>, and 102<sub>4</sub>) for four respective web sites and four user terminals or computers (106<sub>1</sub>, 106<sub>2</sub>, 106<sub>3</sub>, and 106<sub>4</sub>). Connected to Intranet 106 are four servers (102<sub>1</sub>, 102<sub>2</sub>, 102<sub>3</sub>, and 102<sub>4</sub>) and a data processing computer 108. Connected to data processing computer 108 is a data warehouse 116.

[0026] It should be noted that, in describing the present invention, figure 1 shows that only four servers and four user computers are connected to Internet 105. In reality, Internet 105 connects thousands of servers and user computers.

**[0027]** Each of the four servers (102.<sub>1</sub>, 102.<sub>2</sub>, 102.<sub>3</sub>, or 102.<sub>4</sub>) includes a respective web page repository (103.<sub>1</sub>, 103.<sub>2</sub>, 103.<sub>3</sub>, or 103.<sub>4</sub>) and a respective set of server logs (104.<sub>1</sub>, 104.<sub>2</sub>, 104.<sub>3</sub>, or 104.<sub>4</sub>). Each of the four web page repositories (103.<sub>1</sub>, 103.<sub>2</sub>, 103.<sub>3</sub>, or 103.<sub>4</sub>) stores a plurality of web page files (such as HTML, SHTML, DHTML, or CGI files). A web page file may contain links to other type files (such as AVI, GIF, JPEG, and PNG files). (Note: HTML stands for Hypertext Markup Language, SHTML for Secure HTML, DHTML for Dynamic HTML, CGI for Common Gateway Interface, GIF for Graphics Interchange Format, JPEG for Joint Photographic Expert Group, AVI for Audio Video Interleave, and PNG for Portable Network Graphic). The other type files are also stored in one of the four servers. Each of the four set of server logs (104.<sub>1</sub>, 104.<sub>2</sub>, 104.<sub>3</sub>, or 104.<sub>4</sub>) contains access status data (or Internet data) indicating access status for the files that have been accessed, or attempted to be accessed.

**[0028]** Each of the four user computers (106.<sub>1</sub>, 106.<sub>2</sub>, 106.<sub>3</sub>, or 106.<sub>4</sub>) runs a respective web browser (108.<sub>1</sub>, 108.<sub>2</sub>, 108.<sub>3</sub>, or 108.<sub>4</sub>), each of which is able to obtain files from any one of the four servers via Internet 105, and displays these files in a web page format. To obtain a web page file from a server, a web browser sends an Get request to that server. A Get request contains the IP address identifying the user computer on which the browser is being run and a URL (Uniform Resource Locator). The URL contains the name of and path to the web page file. Upon receiving the Get request, the server retrieves the web page file according to the URL in the Get request and sends the web page file to the user computer (on which the browser is being run) identified by the IP address in the Get request. The server then records access status data for the web page file in a server log. Upon receiving the web page file, the web browser displays it as a web page. If the web page file also contains links to other type files, the browser further sends Get requests to the server, so that these files can be obtained and displayed together with the web page file. The links embedded in the web page file contain the names of and paths to these files. After sending these files to the browser, the server records access status data for these files in the server log. If the web page file further contains a link to another web page file, in response to clicking (activating) the link, the browser sends a Get request to the server, so that the web page file can be obtained and a new web page can be displayed. This link contains the name of and path to the web page file. After sending this web page file to the user computer (on which the browser is being run), the server records access status data for the web page file in the server log.

**[0029]** It should be noted that in figure 1 browsers (108.<sub>1</sub>, 108.<sub>2</sub>, 108.<sub>3</sub>, and 108.<sub>4</sub>) interact with servers (102.<sub>1</sub>, 102.<sub>2</sub>, 102.<sub>3</sub>, and 102.<sub>4</sub>) in a stateless fashion. The web browsers (108.<sub>1</sub>, 108.<sub>2</sub>, 108.<sub>3</sub>, and 108.<sub>4</sub>) send requests to servers (102.<sub>1</sub>, 102.<sub>2</sub>, 102.<sub>3</sub>, and 102.<sub>4</sub>) in a

random manner. While a browser (108.<sub>1</sub>, 108.<sub>2</sub>, 108.<sub>3</sub>, or 108.<sub>4</sub>) is sending a sequence of requests to a server (102.<sub>1</sub>, 102.<sub>2</sub>, 102.<sub>3</sub>, or 102.<sub>4</sub>), it does not maintain a constant connection to that server between any two consecutive requests. To a server, it has no control over the sequences of requests; a subsequent request may not have any logical relationship with the previous one; a sequence of requests may come from different web browsers; a request may be generated from a link embedded in a web page file. Consequently, the Internet data are kept as a set of separate and non-correlated data records that are chronologically generated according to the times at which the requests were received and processed. Thus, the Internet data stored in the four sets of server logs (104.<sub>1</sub>, 104.<sub>2</sub>, 104.<sub>3</sub>, and 104.<sub>4</sub>), without further processing, are not meaningful to management and business operation.

**[0030]** As shown in figure 1, data processing computer 108 contains a utility application 112, a page map 114, and a loading utility 115. Via Intranet 107, utility application 112 is able to get access to the four sets of server logs (104.<sub>1</sub>, 104.<sub>2</sub>, 104.<sub>3</sub>, and 104.<sub>4</sub>), to collect data from them, to process the data collected, and to store the processed data in page map 114. Loading utility 115 is able to load the data from page map 114 to data warehouse 116 for further processing.

**[0031]** Referring to figure 2, there is shown a portion of a web page 200, which is associated with a web page file (HTML, SHTML, DHTML, or CGI file) 201.

**[0032]** As shown in figure 2, the portion of web page 200 contains six regions, including: a text region 202; a graphic region 204, which is associated with a link 205 to a GIF file; a graphic region 206, which is associated with a link 207 to a JPEG file; a multimedia region 208, which is associated with a link 209 to an AVI file; a region 214, which is associated with link 215 to other portions of web page 200; and a region 216, which is associated with a link 217 to another web page file. Links 205, 207, 209, 215 and 217 are embedded in web page file 201.

**[0033]** Referring to figure 3, there is shown a plurality of exemplary data records in server logs (104.<sub>1</sub>, 104.<sub>2</sub>, 104.<sub>3</sub>, or 104.<sub>4</sub>) in some detail.

**[0034]** As shown in figure 3, four records J<sub>1-4</sub> indicate the access status for web page file 201 and the other type files (GIF, JPEG and AVI files) that are linked in web page file 201. To better describe the process of generating the four records (J<sub>1-4</sub>), it is assumed that: (1) web page file 201 is stored in page repository 102.<sub>1</sub>, (2) web page file 201 has been accessed by browser 108.<sub>1</sub>, (3) server 102.<sub>1</sub> generates records J<sub>1-4</sub> in server logs 104.<sub>1</sub>, and (4) the four browsers (102.<sub>1</sub>, 102.<sub>2</sub>, 102.<sub>3</sub>, and 102.<sub>4</sub>) are all sending Get requests to server 102.<sub>1</sub>.

**[0035]** To obtain web page file 201, browser 108.<sub>1</sub> sends a Get request to server 102.<sub>1</sub> via Internet 105. The Get request contains the IP address assigned to user computer 106.<sub>1</sub> and an URL indicating the name of and path to web page file 201. Upon receiving the Get request, server 102.<sub>1</sub> retrieves web page file 201 from

web page repository 104.<sub>1</sub> and sends it, via Internet 105, to user computer 106.<sub>1</sub> according to the IP address contained in the Get request. In the meantime, server 102.<sub>1</sub> stores information indicating access status for web page file 201 into record J<sub>1</sub>. Since links 205, 207, and 209 are embedded in web page file 201 to link GIF, JPEG and AVI files respectively, web browser 108.<sub>1</sub> further sends three Get requests to server 102.<sub>1</sub>. Links 205, 207 and 209 contains the file names of and paths to GIF, JPEG, and AVI files, respectively. In addition to containing the IP address assigned to user computer 106.<sub>1</sub>, the three Get requests contain the file names of and paths to the GIF, JPEG, and AVI files, respectively. Upon receiving the three Get requests, server 102.<sub>1</sub> retrieves the GIF, JPEG and AVI files from web page repository 104.<sub>1</sub> and sends them, via Internet 105, to user computer 106.<sub>1</sub> according to the IP address contained in the Get request. In the meantime, server 102.<sub>1</sub> stores information indicating access status for the GIF, JPEG, and AVI files into records J<sub>2</sub>, J<sub>3</sub>, and J<sub>4</sub>, respectively. As shown in figure 2, data records J<sub>1-4</sub> are scattered among the other records in the server logs 104.<sub>1</sub>, because the four browsers (102.<sub>1</sub>, 102.<sub>2</sub>, 102.<sub>3</sub>, and 102.<sub>4</sub>) are all sending Get requests to server 102.<sub>1</sub>, and data records in server logs 104.<sub>1</sub> are chronologically generated according to the times when Get requests have been received and processed by server 102.<sub>1</sub>. It should be noted that, even though figure 3 depicts a process of generating access status information for web page file 210 having a particular web page layout, the principle of figure 3 applies to any web page files having any web page layouts.

[0036] Typically, each of the records in server logs (104.<sub>1</sub>, 104.<sub>2</sub>, 104.<sub>3</sub>, and 104.<sub>4</sub>) contains the following fields:

- IP address assigned to the user computer,
- user's domain name,
- name of the request (such as Get),
- time stamp on which the request was received,
- URL (including access path to the file and parameters passed),
- server name,
- IP address of the server,
- bytes received from the browser,
- bytes sent to the browser, and
- status code indicating operational status of processing the request.

[0037] Referring to figure 4, there is shown a flowchart illustrating the operation of forming page map 114 by utility application 112 shown in figure 1, in accordance with the present invention.

[0038] In step 402, utility application 112 collects Internet data stored in server logs (104.<sub>1</sub>, 104.<sub>2</sub>, 104.<sub>3</sub>, and 104.<sub>4</sub>) via Intranet 107.

[0039] In step 404, utility application 112 identifies what types of servers that have generated the Internet data, because the four sets of server logs (104.<sub>1</sub>, 104.<sub>2</sub>,

104.<sub>3</sub>, and 104.<sub>4</sub>) can be generated by different types of servers. For example, the four servers (102.<sub>1</sub>, 102.<sub>2</sub>, 102.<sub>3</sub>, and 102.<sub>4</sub>) shown in figure 1 can be a web server, hosting web server with virtual domains, commerce server, and proxy server, respectively. Since different types of servers may generate Internet data with different formats, the data format and content in one set of server logs (104.<sub>1</sub>, 104.<sub>2</sub>, 104.<sub>3</sub>, or 104.<sub>4</sub>) may be different from those in the other three sets of server logs. By identifying server type, utility application 112 can process the Internet data in a way that is suitable to the data format and content in the identified server logs. In doing so, utility application 112 can process and combine Internet data generated by different types of servers. In the present invention, the server type can be identified by the fields included and orders of the fields in the server logs.

[0040] In step 406, utility application 112 removes non-useful data from the data collected in step 402. By way of example, a backspace in a URL is non-useful character; one of the two "/" in a URL is a non-useful character because two "/" have the same meaning as one "/" to a server. Thus, the backspace and one "/" can be removed. By way of another example, the data in a record for retrieving a file associated to a unrecognizable URL is not useful, because no file can be found in response to the URL. Thus, the whole record can be removed. Typically, status code field in a data record indicates whether a request has been successfully processed or not. This step is advantageous because server logs may contain a huge volume of data. Keeping non-useful data in applications, such as data warehouse applications, not only is wasteful of storage space, it may also cause errors in the reports and during analysis.

[0041] In step 408, utility application 112 identifies records that store data indicating file access status for web page files (HTML, SHTML, DHTML, or CGI files). In the example shown in figure 3, record J<sub>1</sub> for web page file 201 shown in figure 2 will be identified in step 408.

[0042] In step 410, utility application 112 identifies records that store data indicating file access status for other type files (such as GIF, JPEG and AVI files) that are linked into respective web page files. In the example shown figure 3, records J<sub>2-3</sub> can be identified to be linked to web page file 201 shown in figure 2.

[0043] In step 412, utility application 112 correlates the records for the identified other type files with their respective identified web page files by using the IP address (assigned to the user computer running the browser) and time stamp fields in the these records. As described above, if any other type files are linked into a web page file after a browser has received the web page file, the browser immediately sends requests out to retrieve the other type files. Hence, the IP address in the request for retrieving the web page file is the same IP address in the requests for retrieving the other type files. Also the time at which the request for retrieving the web page file was received should be close to those at which

the requests for retrieving the other type files were received. Therefore, utility application 112 correlates the following records together:

- (1) a particular record for a particular web page file, which contains an IP address and time stamp, and
- (2) a set of records for the other type files, which contain the same IP address with that in the particular record; and contain the time stamps close to (within one or two seconds, for example) that in the particular record.

In the example shown in figure 3, records J<sub>2-4</sub> can be correlated with record J<sub>1</sub>.

**[0044]** In step 414, for each of the web page files, utility application 112 calculates a length by combining the bytes sent for the one web page file with the bytes sent for the other type files linked in the one web page file. In the example shown in figure 2, the bytes sent for web page file 201 will be combined with the bytes sent for GIF, JPEG and AVI files. The length is useful for an Internet Service Provider to manage its operation, because it can provide the information to determine the bandwidth used and the cost to send these files.

**[0045]** In step 416, utility application 112 stores the data processed in the steps (406, 408, 410, 412, and 414) in page map 114 shown in figure 1.

**[0046]** Referring to figure 5, there is shown a plurality of exemplary data records in page map 114, in accordance with the present invention.

**[0047]** As shown in figure 5, page map 114 contains a plurality of data records 502<sub>1</sub>, 502<sub>2</sub>, ..., 502<sub>j</sub>, .... Each of the records may include several physical or logical storage units. Each of the records stores the IP address used by a browser to retrieve a web page file, and the correlated information indicating the access status for the web page file and other type files linked to the web page file. Each of the records also stores a combined length for all the bytes sent for the web page file and the other type files.

**[0048]** Referring to figure 6, there is shown an exemplary computer system 600 used as data processing computer to run utility application 112, in accordance with the present invention.

**[0049]** As shown in figure 6, computer system 600 comprises a processing unit 602, a memory device 604, a hard disk 606, a disk drive interface 608, a display monitor 610, and display interface 612, a bus interface 624, a mouse 625, a keyboard 626, a network communication interface 634, and a system bus 614.

**[0050]** Hard disk 606 is coupled to disk drive interface 608, display monitor 610 is coupled to display interface 612, and mouse 625 and keyboard 626 are coupled to bus interface 624. Coupled to system bus 614 are: processing unit 602, memory device 604, disk drive interface 608, display interface 612, bus interface 624, and network communication interface 634.

**[0051]** Memory device 604 is able to store programs

(including instructions and data). Operating together with disk drive interface 608, hard disk 606 is also able to store programs. However, memory device 604 has faster access speed than hard disk 606, while hard disk 606 has higher capacity than memory device 604.

**[0052]** Operating together with display interface 612, display monitor 610 is able to provide visual interface between programs being executed and a user.

**[0053]** Operating together with bus interface 624, mouse 625 and keyboard 626 are able to provide inputs to computer system 600.

**[0054]** Network communication interface 634 is able to provide an interface between computer system 600 and Intranet 107.

**[0055]** Processing unit 602, which may include one or more processors, has access to memory device 604 and hard disk 606, and is able to control operations of the computer by executing programs stored in memory device 604 or hard disk 606. Processing unit 602 is also able to control the transmissions of programs and data between memory device 604 and hard disk 606.

**[0056]** In the present invention, utility application 112 can be stored in either memory device 604 or hard disk 606, and be executed by processing unit 602.

**[0057]** While the invention has been illustrated and described in detail in the drawing and foregoing description, it should be understood that the invention may be implemented through alternative embodiments within the spirit of the present invention. Thus, the scope of the invention is not intended to be limited to the illustration and description in this specification, but is to be defined by the appended claims.

## Claims

1. For use with a server that contains a plurality of web pages and logs, each of the web pages containing a web page file and one or more other type files, each of the logs containing data for indicating access status web page files and the other type files, a method comprising the steps of:

receiving data from the server;  
identifying data for a web page file;  
identifying data for other type files that are linked in said web page file; and  
correlating said data for said other type files with said data for said web page file.

2. The method of claim 1, said web page file, or each of said other type files linked in said web page file, having a file length, and the method further comprising the step of:  
combining file lengths of said web page file and said other type files that are linked in said web page file.

3. The method of claim 1 or claim 2, further comprising the step of:

filtering non-usable character(s) from said data for indicating access status of said web page file and said data for indicating access status of said other type files.

4. For use with a server that contains a plurality of web pages and logs, each of the web pages containing a web page file and one or more other type files, each of the logs containing data for indicating access status of web page files and other type files, a method comprising the steps of:

receiving data from the server;  
identifying data for web page files that have been accessed;  
identifying data for other type files that are respectively linked in said accessed web page files; and  
correlating said data for said other files that are respectively linked in said accessed web page files into said data for said accessed web page files.

5. The method of claim 4, further comprising the step of:

filtering non-usable character(s) from said data for indicating access status of said accessed web page files and said data for indicating access status of said accessed other type files.

6. The method of claim 4 or claim 5, the web pages being accessed by different types of users, the method further comprising the step of:

sorting said correlated data in accordance with the different types of users.

7. The method of claim 4, claim 5 or claim 6, each one of said web page files, or each of said other type files having a file size, and the method further comprising the step of:

combining file sizes of said other type files linked in respective web page files.

8. For use with a plurality of servers with each of them containing a plurality of web pages and logs, each of the web pages containing a web page file and one or more other type of files, each of the logs storing data for indicating access status of web page files or other type files, a method comprising the steps of:

receiving data from any one of the servers;  
identifying a server type for said one server;  
identifying data for the web page files that are stored in said one server and have been accessed;

identifying formation for other type files that are linked in said respective accessed web page files stored in said one server; and  
correlating said data for said other type files that are linked in said respective accessed web page files with said data for said accessed web page files stored in said one server.

9. The method of claim 8, each of said web page files and each of said other type files having a file size, and the method further comprising the step of:

combining file sizes of said files linked in said respective accessed web page.

10. The method of claim claim 8 or claim 9, further comprising the step of:

filtering non-usable character(s) from said identified data.

11. The method of claim 8, claim 9 or claim 10 the accessed web page files and other type files being accessed by different regions, the method further comprising the step of:

sorting said correlated data in accordance with the different regions.

12. The method of any one of the preceding claims,

wherein said web page file is an HTML file, a DHTML file, an SHTML file, or a CGI file; and wherein said other type files are GIF files, JPEG files, or AVI files.

13. The method of any one of the preceding claims, further comprising the step of:

storing said correlated data into an output file.

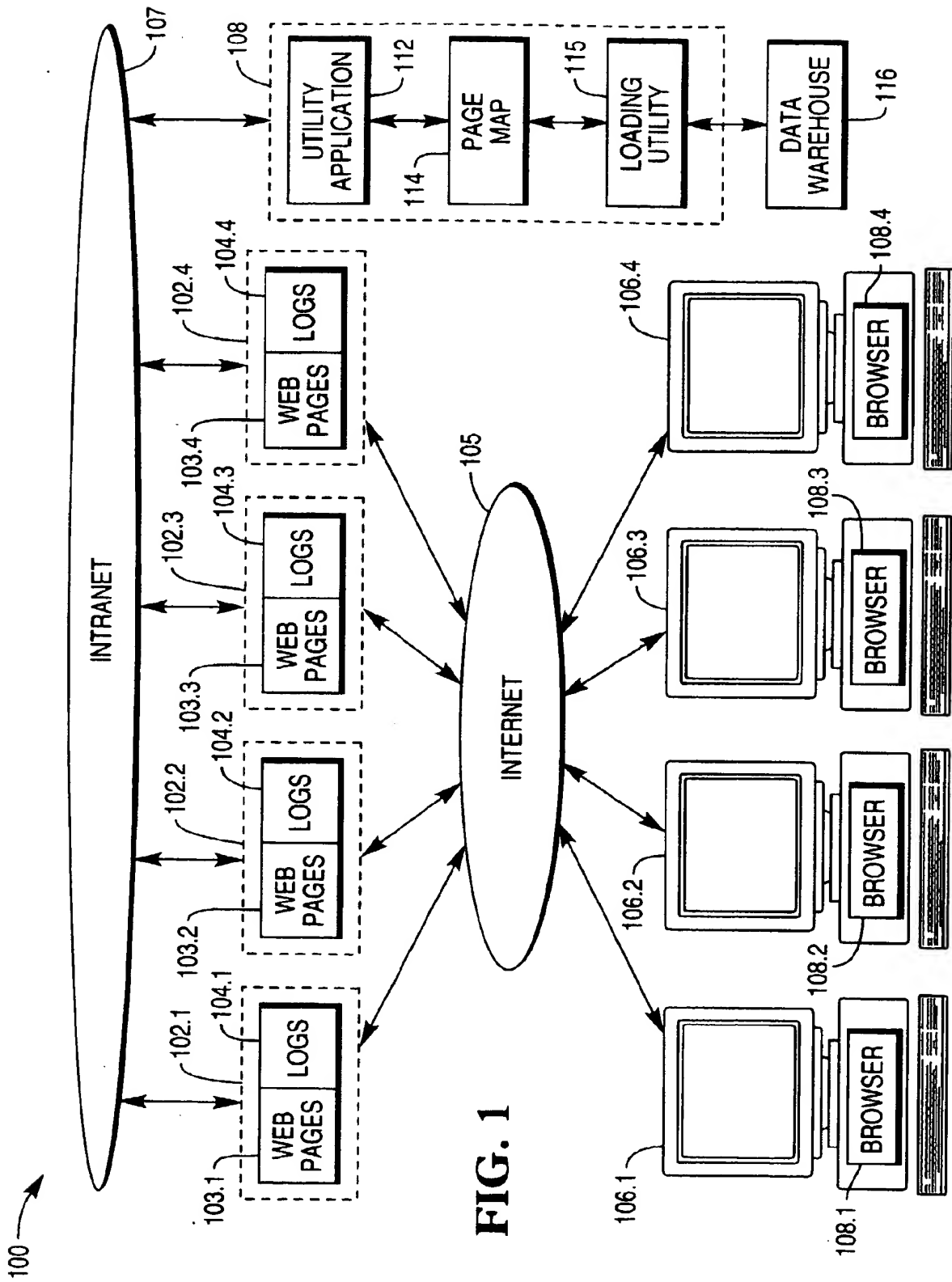
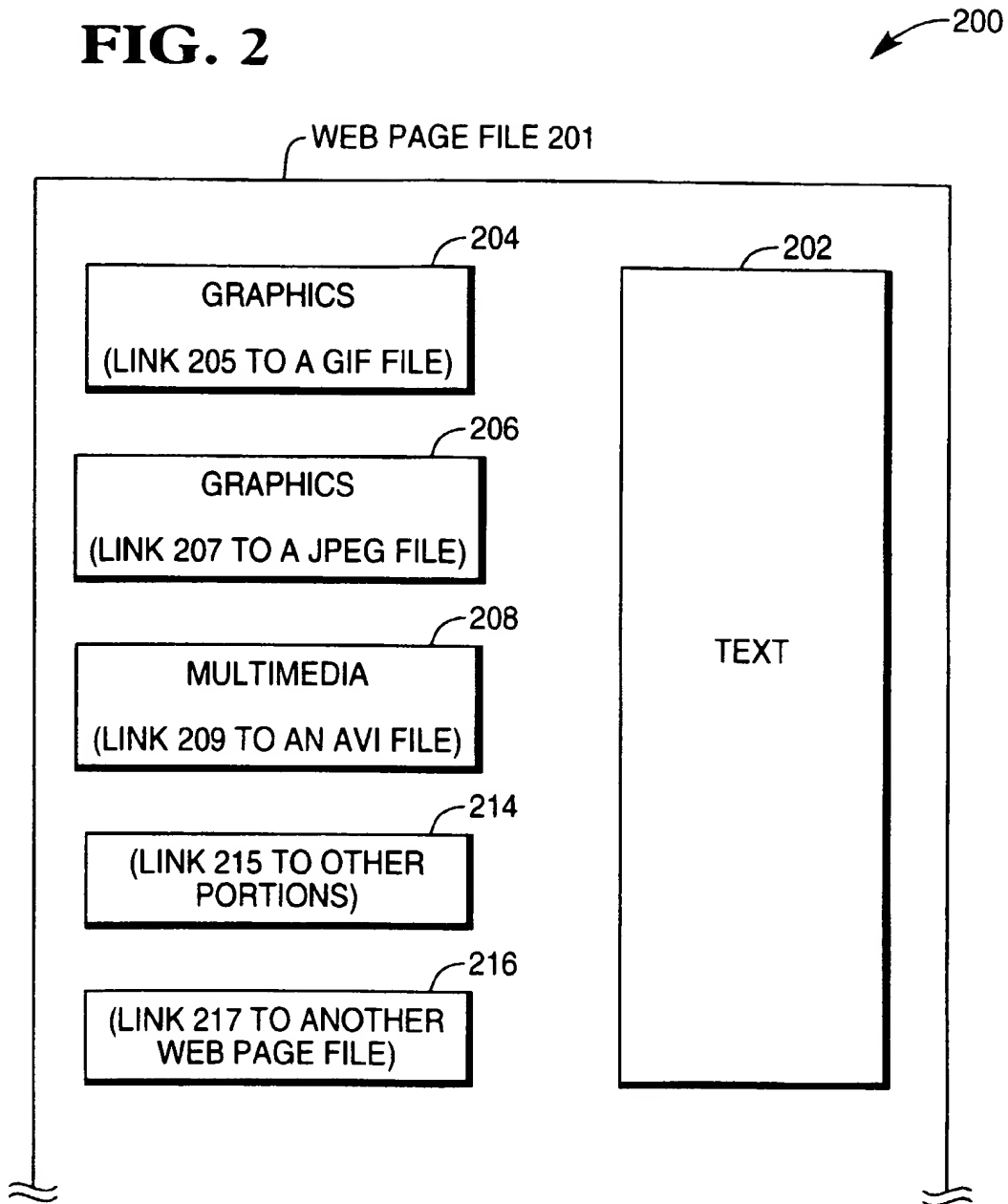


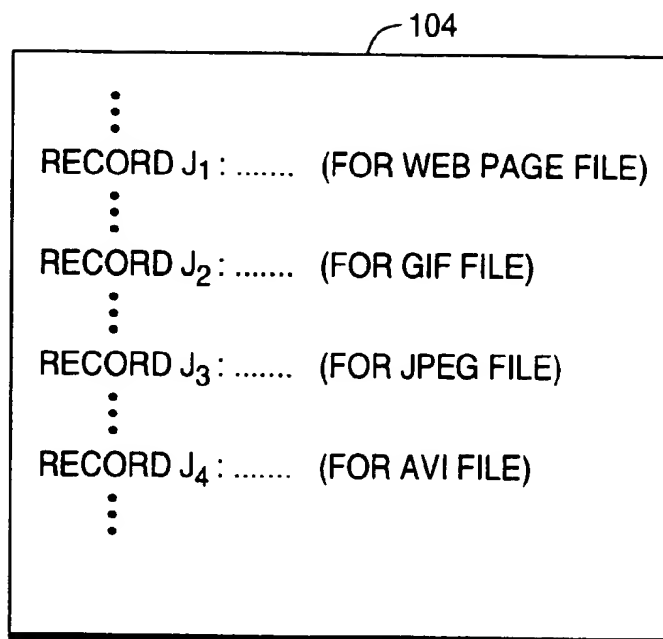
FIG. 1

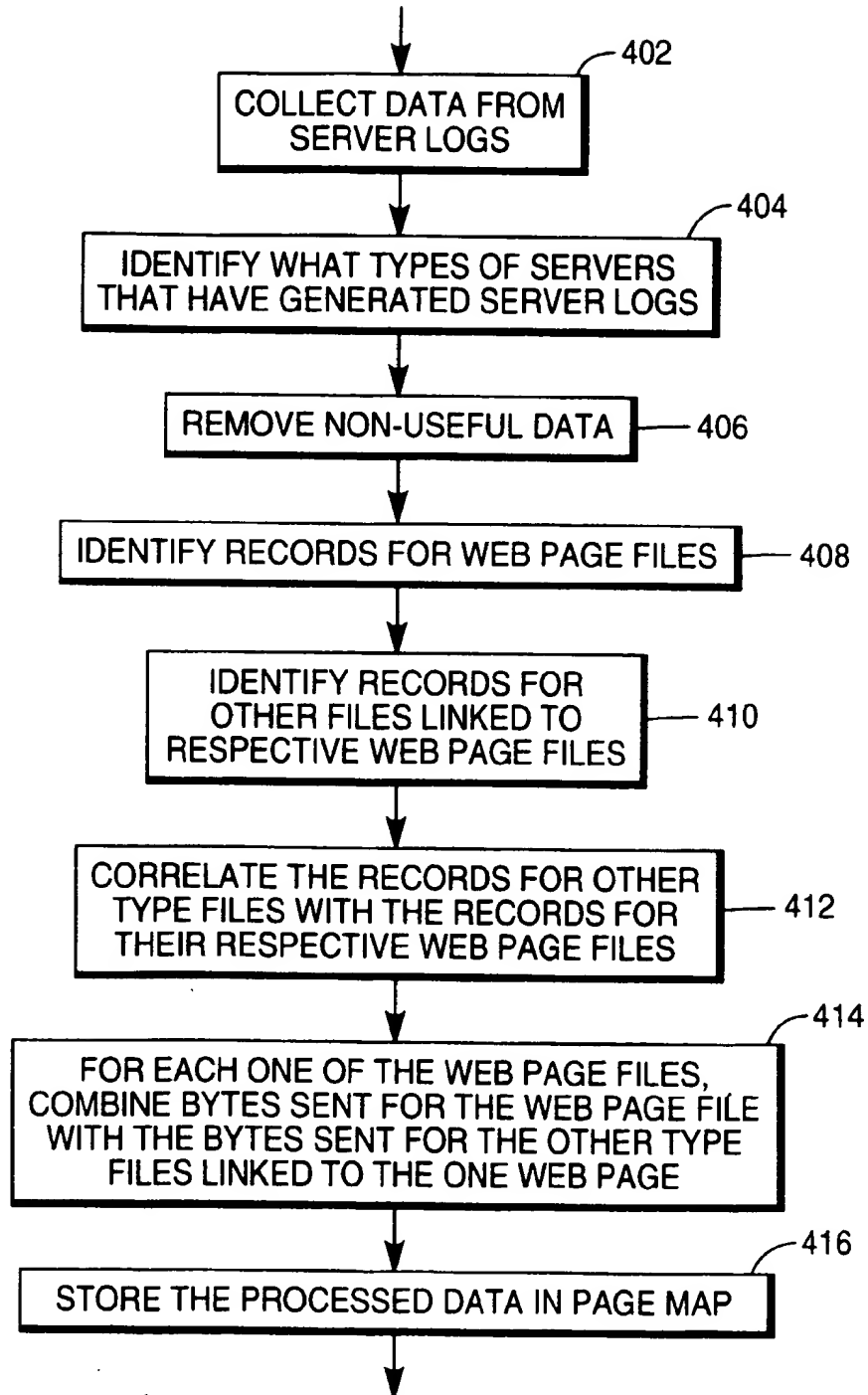


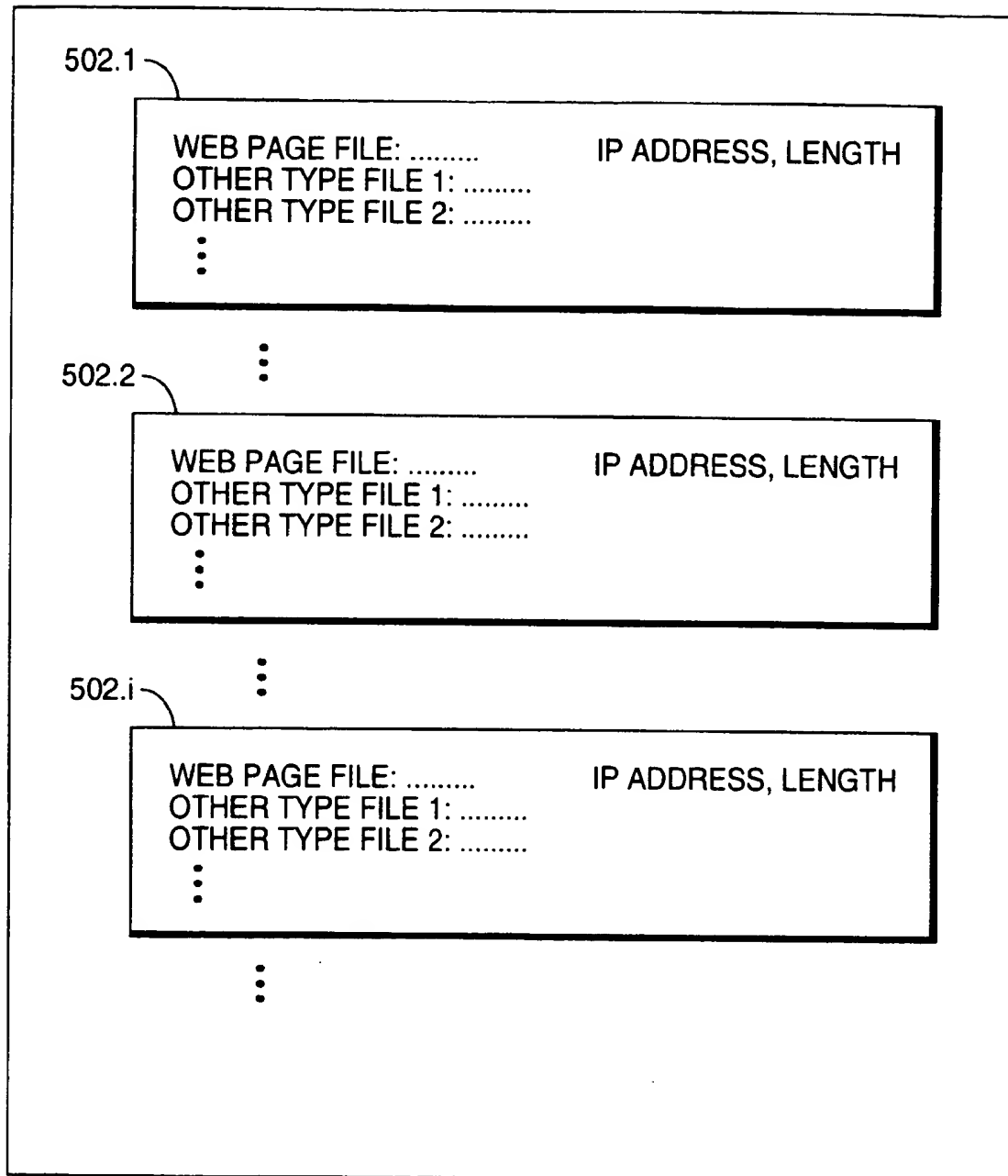
**FIG. 2**



**FIG. 3**



**FIG. 4**

**FIG. 5**

**FIG. 6**